Introduction to CIAO

(Chandra Interactive Analysis of Observations)

From "ahelp ciao"....

- A powerful data analysis system originally written for the needs of users of the Chandra X-ray Observatory.
- Built to handle *N-dimensional* data without concern about which particular axes are being analyzed: Chandra is the first mission with 4-dimensional data (2 spatial, time, energy) in which each dimension has many independent elements.
- Mission independent (a part from a few instrument specific tools).

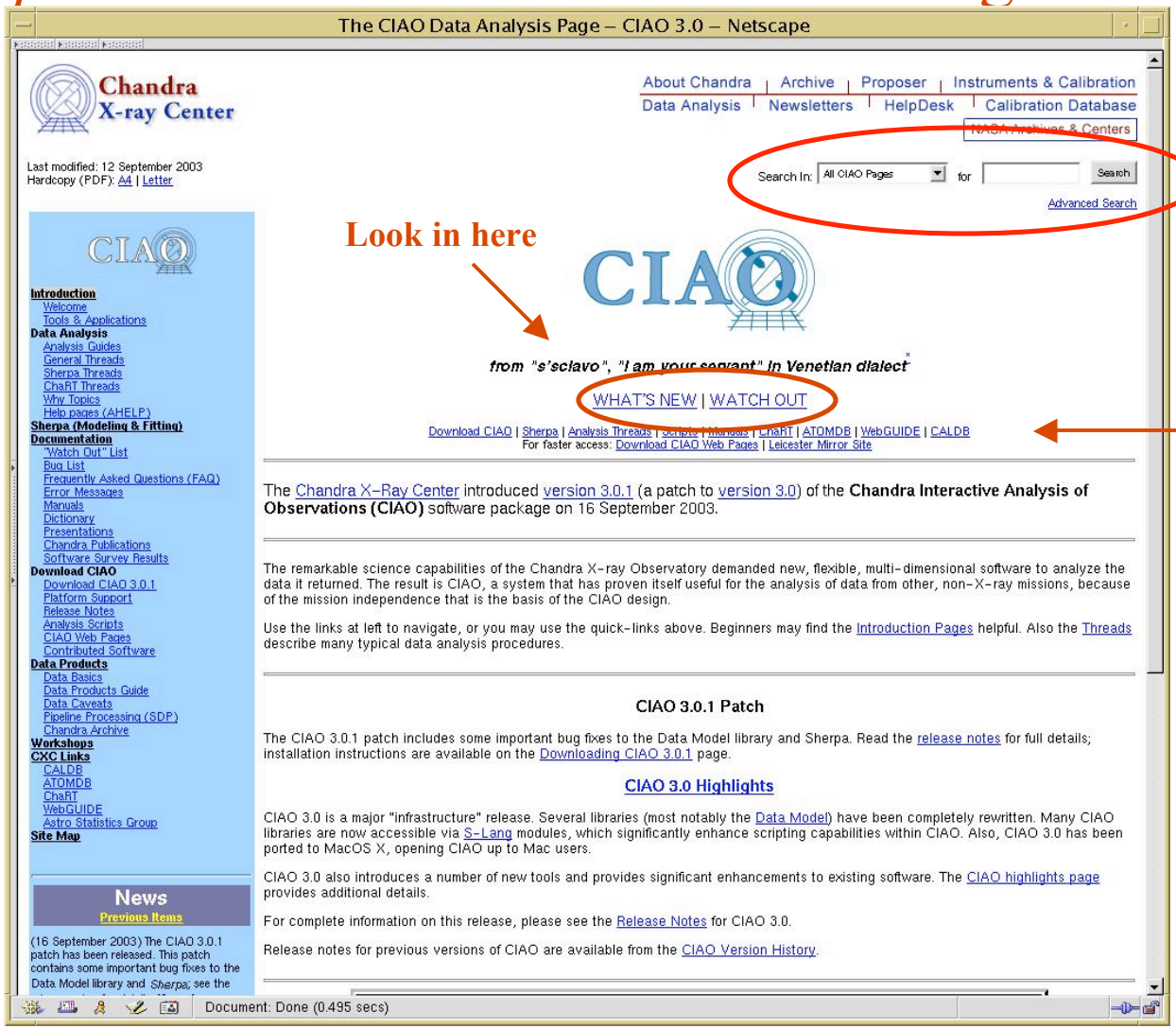
- Provides users with the ability to filter down and project the 4-D Chandra event data to manageable sizes and convenient array: all CIAO tools take a '*filtering and binning*' specification on the command line, making use of a general purpose 'regions' syntax: "`ahelp dm`" for information on the Data Model that makes all this possible.

- To keep track of how the data had been filtered and binned CIAO relies on the 'data subspace'. Tools keep track of this subspace automatically and allow users to review previous data processing: see "`ahelp subspace`".

- The CIAO design allows close interconnection of tools. For example, the output of any of source detection program can be fed into *dmextract* to create a summed spectrum which can then be fit in *Sherpa*.

- The modeling and fitting tool ***Sherpa*** is central to the CIAO system. Sherpa performs forward fitting of models to data in N-dimensions. Sherpa includes the "**S-Lang**" language which can be used for scripting and data manipulation. GUIDE links Sherpa results to the APEC/APED plasma database, enabling the identification of spectral lines and the use of their properties in further fitting. See "`ahelp sherpa`", "`ahelp slang`", "`ahelp guide`".

http://cxc.harvard.edu/ciao : it all begins here!



CIAO help!

AHELP

- CIAO comes with its own help system called **ahelp**.
- Every component of the CIAO system has its own help text: tools (e.g. `dmcopy, wavdetect`), applications and their components (e.g. `sherpa, fit, model`), scripting language and its functions (e.g `slang, get_fit, get_data`) and numerous concepts (e.g. `regions, coords, dmsyntax, etc.`).
- The entire collection of ahelp files currently has about 900 pages!

The **ahelp** commands (see `ahelp ahelp` for details) accesses the CIAO on-line documentation. For example:

% about <word> (e.g. about contour) gives a list of subjects related to the query
% ahelp <name> (e.g. ahelp tgextract) provides by default the ASCII version of the help file
% ahelp -w <name> loads into a browser the HTML version supplied with the CIAO distribution
% ahelp -i <name> access the CIAO site which contains HTML and PDF versions of the help files (often more up-to-date than the released ones).

DATA ANALYSIS GUIDES : *start from here!*

- A roadmap through the threads.
- Arranged by instrument (ACIS, HRC) or type of analysis (e.g. extended sources)

DATA ANALYSIS THREADS (*General, Sherpa, Chart*)

- About 100 CIAO processing recipes designed to teach users by leading step-by-step through a procedure.
- More added weekly (look for the "new" or "updated" icons!).
- Several threads and groups of threads have been or will be made more automated in the form of scripts.
- PDF version available on-line.
- New layout for CIA3.0 with quick "overview" (synopsis, purpose, etc.)

MANUALS

CIAO manuals are currently being written or updated: major updates were done to the DM (Data Model/Data Manipulation) and Detect manuals. Updated Sherpa and Chips manuals are scheduled to be released in the next few weeks.

CIAO WEB PAGES

Read the *What'sNew, Watchout, Bugs, Caveats, Dictionary, FAQ pages !*

HELPDESK

When everything else fails....

http://cxc.harvard.edu/helpdesk/

with CIAO version, platform, data, purpose etc…

WHAT IS CIAO?

- Collection of programs (*tools*, *applications, scripts, S-Lang modules*).
- Generally run from the shell; some part can be run from GUIs.
- Source code is available.
- Available on several platforms; currently supported:
  - Solaris 8
  - Redhat Linux 6.2 (also tested on Suze 7, Slackware 7.1, Redhat 7.1)
  - Redhat Linux 8 (also tested on Redhat 9)
  - Mac OS X 10.2.6 (***new in CIAO3.0***)
  - Compaq Alpha/True64 Unix 5.1A

The CIAO Environment

We recommend the use of an alias called "`ciao`" to start up the system: it sets up number of environment variables and path assignments.

`ciao -v` tell what version you are using (useful when reporting problems)

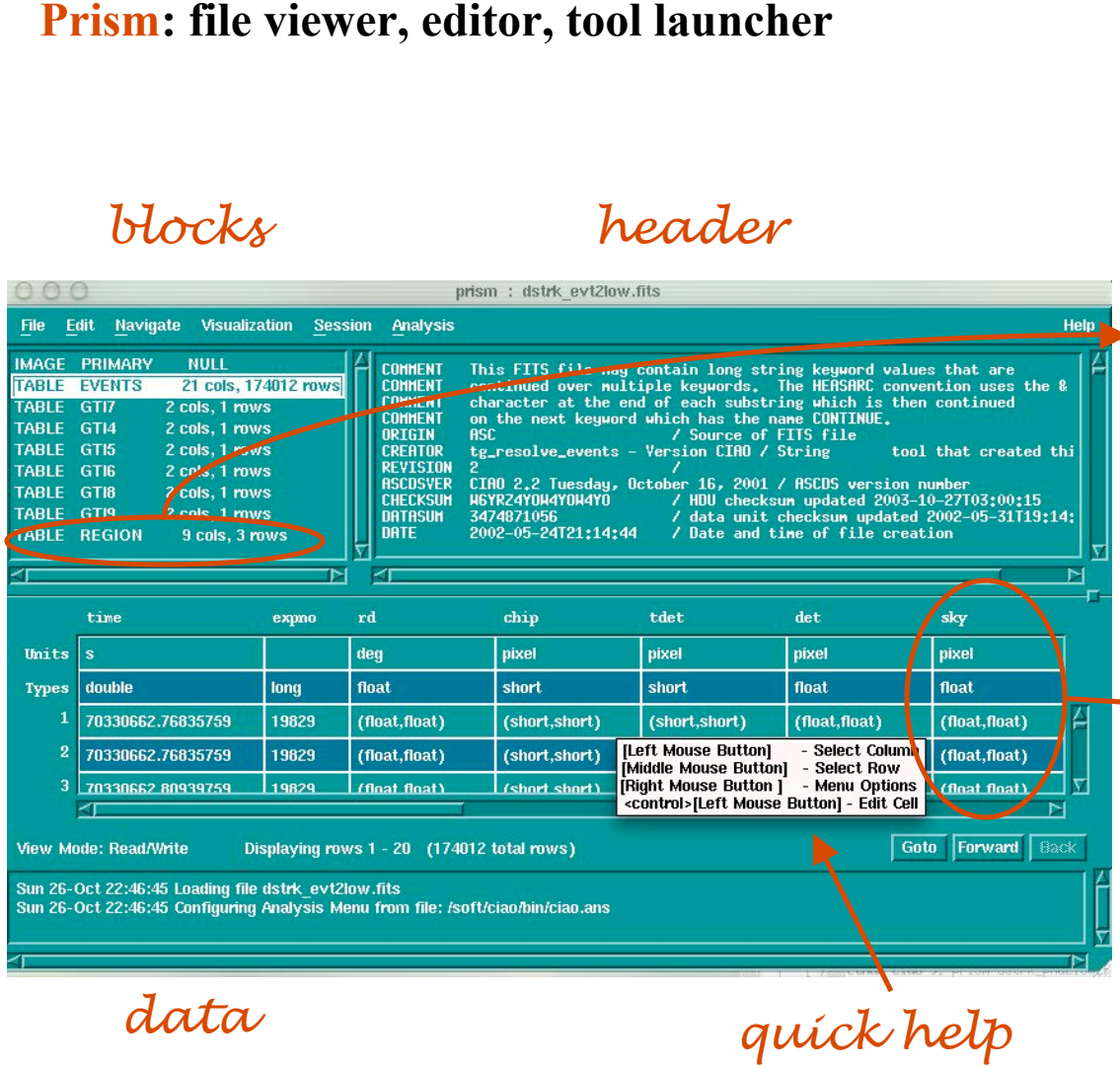
`ciao -h` help on how to setup CIAO

General Concepts

File Format

- Chandra data is stored in FITS format. ASCII files can be handled by many tools. Support for the IRAF format (IMH and QPOEs) is limited.
- CIAO stores processing state/information along with data (keywords, subspace).
- A single file can contain multiple "datasets" (e.g. data, GTI, weight map, regions) stored in "blocks".
- Blocks can contain image or table data. Table columns can be vectors.
- dmlist (a command line tool) or prism (a GUI) are available to view file contents.

Prism: file viewer, editor, tool launcher



Parameter Files (`ahelp parameter`)

- Processing parameters for CIAO tools can be set on the command line or, as with IRAF and FTOOLS, using parameter files.
- Parameters files are stored in *$HOME/cxcds_param/* by default, are called *<tool>.par* (e.g. dmcopy.par), and are ASCII files.
- A "Parameter Editor GUI" (peg - *new in CIAO3.0*) and a number of routines (e.g. plist, pset, punlearn) are provided to read and write to these files. A S-Lang interface to the CXC parameter library is now also available (see `ahelp paramio`).

Peg in action:

```
unix% peg dmlist
```

Filters, Regions, and GTIs

- filtering (removal of unwanted events) is an essential part of X-ray analysis - e.g. to remove periods of high background or poor aspect solution, exclude uninteresting sources from an image etc.
- the DataModel (DM) provides great filtering flexibility: e.g. dmstat "evt2.fits[EVENTS][energy>300][cols -grade]" (see `ahelp filtering`, `ahelp dmimgfiltering`)
- GTIs (Good Time Intervals) are used to define what times periods of the observation can be used (i.e. contain valid data). They are generally stored as a block in the event list (see `ahelp chandra times`)
- Regions are used to define the source and background areas of an image. They are text files that can be created manually or within ds9, and are used as a filter (e.g. "[sky=region(source.reg)]"). (see `ahelp dmregions`)
- Subspace records the filters applied to a file; dmlist can read this history using opt=subspace (see `ahelp subspace`)

CIAO overview

Data manipulation: copy, filter, extraction, stats, etc.

Data preparation (or Chandra-specific instrument tools): update calibration, correct for instrumental effects, find & extract grating data, create aspect histograms

Response tools: exposure map, PSF, RMF and ARF

Source Detection: celldetect, wavdetect, vtpdetect

Timing & Background tools: lighcurve, power specrtrum, barycenter correction

Convolutions, Transforms, & Smoothing: csmooth, aconvolve, acrosscorr, apowerspectrum

Plotting: (*) ChIPS

Modeling/Fitting: (*) Sherpa

Spectral Line Identification: GUIDE, ISIS

S-Lang: modules (parameter, region, group, pixlib (coord. transformation), caldb, stack, xpa, varmio); shell ("slsh") to execute S-Lang scripts on the command line

GUIs: DS9, peg, prism, filtwin

(*) powerful data manipulation and scripting capabilities are now possible with the inclusion of the S-Lang interpreted language.

The "Data Model" and the Data Manipulation Tools

- The CXC analysis and processing software is built on a common versatile interface library called the CXC Data Model (or just DM).
- The DM provides users with a powerful built-in data filtering and binning capability.
- The name "Data Model" reflects the fact that the interface can be used on data files of different format (all described by a single abstract description - the same "model") in a transparent way.
- The DM was completely rewritten in the CIAO3.0 (no change for users; speed somewhat increased but still undergoing improvement)

- As of CIAO3.0 the formats supported by the DM library include FITS, and with some limitations IRAF QPOE and IRAF IMH. Complete ASCII support is on the way.
- An important characteristic of the DM is that ANY program that asks for a data file name as input accepts a *"virtual file"* string which causes the program to see a filtered version of the file in question.
- The "virtual file" syntax is also commonly used to create on disk a filtered version of the input file.
- Another important characteristic of the DM is that all columns of event lists are treated "equally": for example binning is allowed not only in spatial coordinates but also in e.g. time, or energy coordinate, giving the ability of creating multidimensional images in space-energy, or space-time, etc.

Data Manipulation Tools

The four DM "core" tools are:

dmlist: list contents or structure of a file

dmcopy: filter and bin tables and images

dmextract: make a histogram table file (e.g. PHA file, lightcurve file) from a table column. Generate count histogram on supplied regions for a spatial table or image file.

dmgti: create custom Good Time Intervals (GTIs) from a constraint expression

30+ data manipulation tools are included in CIAO3.0

DATA MODEL SYNTAX (`ahelp dmsyntax`)

- All CIAO tools use the DM library and therefore accept as input "virtual files" described using the DM syntax.
- In the DM context a "virtual file" in represented by a filename followed by a series of optional qualifiers in square brackets [ ]:

"filename[block][filter][columns/binning][options][rename]"

where:

block - is the "section" of the file to use

filter- is the filter to be applied

columns/binning - specifies either the columns from a table to be included in an output table or the binning. When binning the data to generate an n- dimensional image, the range and binsize (min:max:bin) must be specified.

options - a sequence describing special options for the DM library

rename - specifies a name for the new block

Note that:

- the order of the qualifiers generally matters, however...
- not all qualifiers need to be present always

Simple examples of "virtual files":

- Select the first three columns of the EVENTS block by number:
  ```
  acisf01843N001_evt2.fits[EVENTS][time=84245787:84247000][cols #1,#2,#3]
  ```
  or by name:
  ```
  acisf01843N001_evt2.fits[EVENTS][grade=0,2,3][cols time,ccd_id,node_id]
  ```
  after filtering in time or grade

- Bin an events file to create a PI spectrum for a specified region (input of dmextract):
  ```
  acisf01843N001_evt2.fits[EVENTS][sky=region(mysrc.reg)][bin pi=1:1024:1]
  ```
  or an image (input of dmcopy):
  ```
  acisf01843N001_evt2.fits[EVENTS][pha<100][bin x=320:480:4,y=320:480:4]
  ```

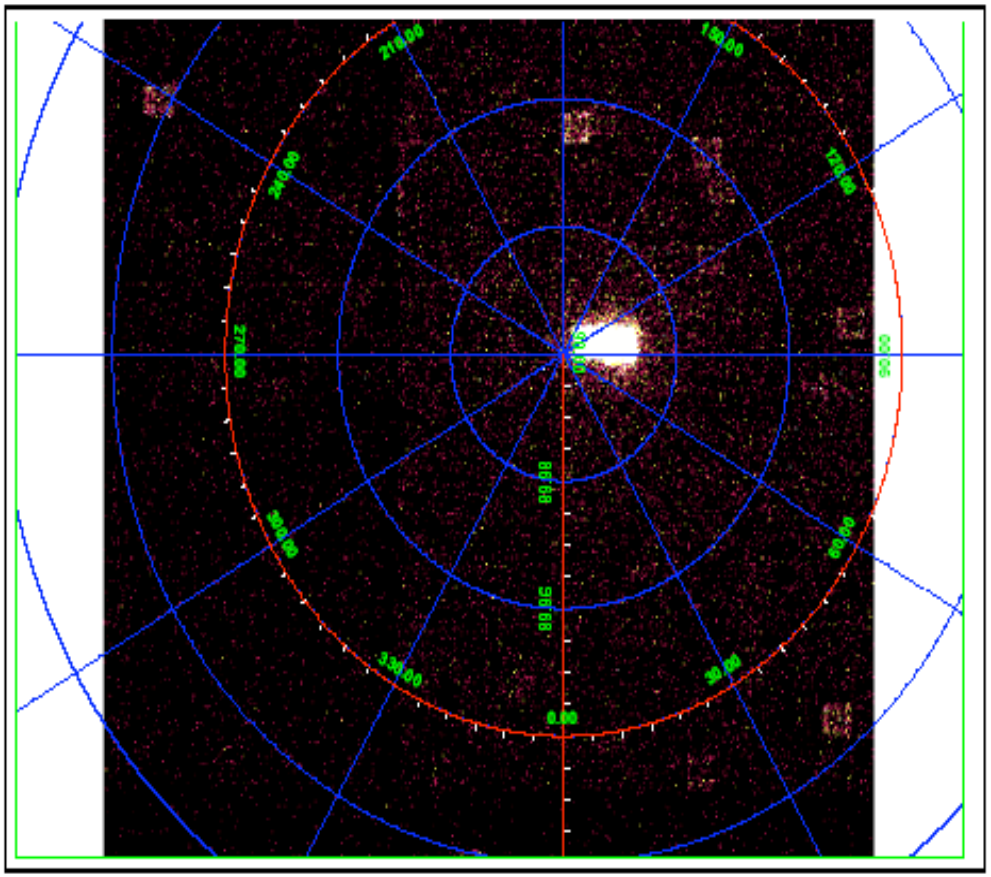
In the examples above:

block: [EVENTS]
filter: [time=84245787:84247000]
[grade=0,2,3]
[sky=region(mysource.reg)]
[pha<1000]
columns/binning:
[cols time,ccd_id,node_id]
[cols #1,#2,#3]
[bin pi=1:1024:1]
[bin x=320:480:4,y=320:480:4]

DM Examples 1: Detector Image

Imaging on multiple coordinate systems: first, let's look at a region in detector coordinates, filtered on energy and time.

dmcopy "merge3e.fits[energy=500:2000, time=:63940080, 63940180:][bin detx=3500:4500:2, dety=3500:4500:2]" det.img



DM Examples 2: Sky Image

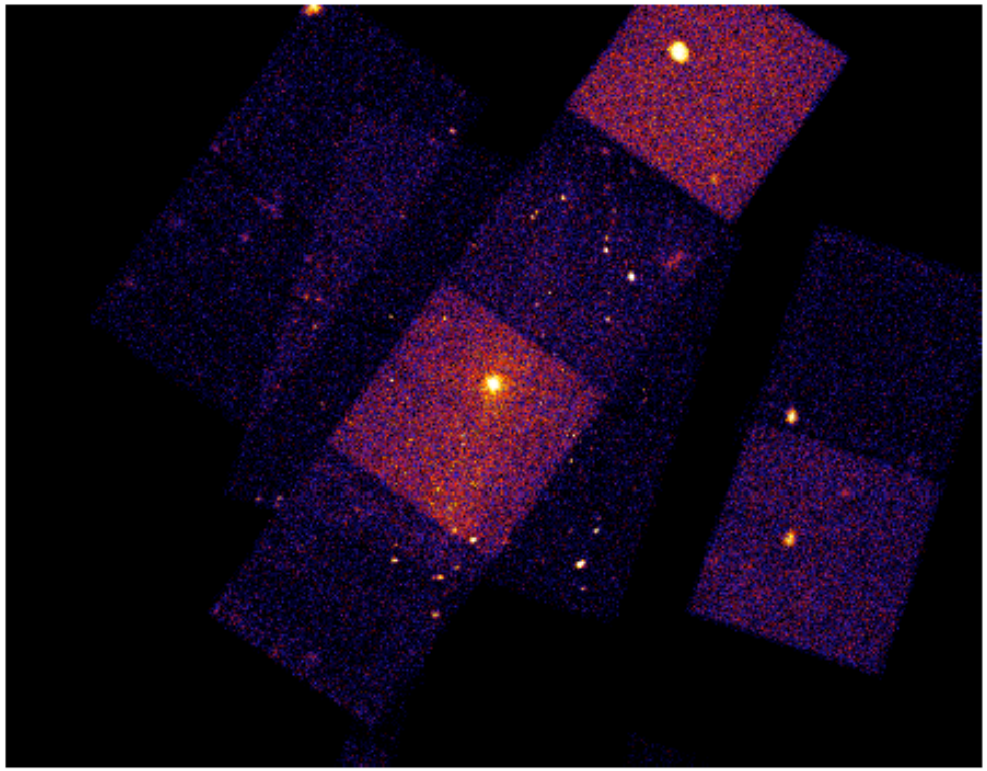
Now look at the same photons but in sky coordinates

dmcopy "merge3e.fits[energy=500:2000, time=:63940080,63940180:,detx=3500:4500,dety=3500:4500][bin x=3200:4800:2, y=3200:4800:2]" sky.img

DM Examples 3: Merged sky image

The whole field was created by merging three separate observations.

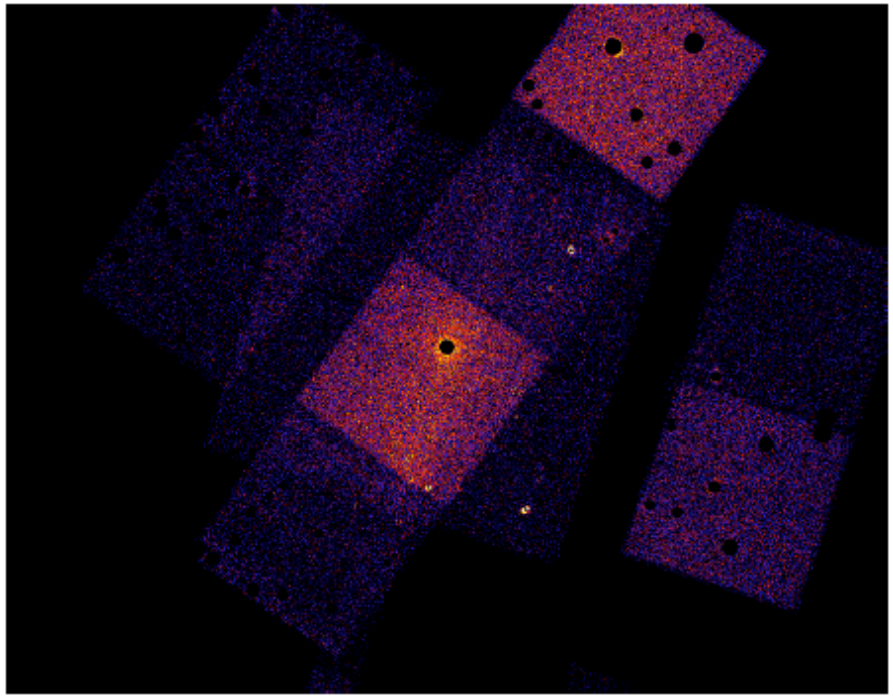
dmmerge “786.fits, 787.fits, 1730.fits” outfile=merge3e.fits



DM Examples 4: Removing sources

We can generate a background image by removing sources found by the automatics source detection program.

dmcopy “merge3e.fits[exclude sky=region (gg.reg)]” exclude.fits



DM Examples 5: Infrared spectroscopy data

ISO data: LWS LSAN file. This is a very simple file by wavelength and flux for the different detectors and scans are mixed together. We can use the DM tools to isolate a single scan and dump wavelength versus flux for it.

dmcopy "lasan59901083.fits[lsancnt=4][cols lsanwav, lsanflx]"
subset.fits

dmlist "lsan59901083.fits[lsanscnt=4][cols lsanwav, lsanflx]"
data,raw outfile=lis.asc